UNIVERSITY OF HELSINKI

FACULTY OF SOCIAL SCIENCES

Effects of Corpus Size on Word Similarity Model

MASTER'S THESIS IN STATISTICS

Joni OKSANEN

10.11.2020

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

Tiedekunta/Osasto — Fakultet/Sektion — Faculty	Laitos — Institution — Department
Faculty of Social Sciences	

Tekijä — Författare — Author
Joni Oksanen

Työn nimi — Arbetets titel — Title

Effects of Corpus Size on Word Similarity Model

Oppiaine — Läroämne — Subject
Statistics

Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages
Master's Thesis	November 2020	41 p.

Tiivistelmä — Referat — Abstract

Text mining methods provide a solution to the task of extracting relevant information from large text datasets. These methods can be applied to extract the relevant parts of Suomi24 internet health discussion to analyze how people discuss and negotiate their health through words, which represents medication or symptoms. Semantic similarities between these two concepts can be examined by learning the word vector representations from data and exploring the vector space using Word2Vec, a popular word embedding method.

This thesis reviews how the training of word similarity models is affected by increasing corpus size using text retrieval methods.The effects of corpus size are examined by comparing the measured cosine similarity distances between word vectors representations in two different vector spaces. Word vector representations are learned using two different sized corpora. The first corpus includes only messages from the health discussion area of Suomi24. The second corpus includes the same messages as the first corpus, but also messages from other discussion areas, which include health related words.

Cosine similarities are evaluated on using concept vocabularies including relevant health related words. Increasing the number of training examples by almost 30 % did not have a drastic effect on the qualities of the training data. The results did not indicate a distinct connection between corpus size and the measured cosine similarity distances between word vector representations of health related words.

Avainsanat — Nyckelord — Keywords
word vector representations, word2vec, suomi24, text mining, information retrieval

Säilytyspaikka — Förvaringsställe — Where deposited

Muita tietoja — Övriga uppgifter — Additional information

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

Tiedekunta/Osasto — Fakultet/Sektion — Faculty	Laitos — Institution — Department
Valtiotieteellinen tiedekunta	

Tekijä — Författare — Author
Joni Oksanen

Työn nimi — Arbetets titel — Title

Effects of Corpus Size on Word Similarity Model

Oppiaine — Läroämne — Subject
Tilastotiede

Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages
Pro Gradu -tutkielma	Marraskuu 2020	41 s.

Tiivistelmä — Referat — Abstract

Tekstilouhinnan menetelmien avulla pystytään erottelemaan ja poimimaan oleellista tietoa suurista tekstiaineistoista. Näitä menetelmiä hyödyntäen voidaan tutkia Suomi24-keskustelupalstan viestejä ja sitä, miten käyttäjät puhuvat omasta terveydestään. Kahden käsitteen välistä semanttista samankaltaisuutta voidaan tarkastella vektoriavaruudessa kouluttamalla Word2Vec-malli, joka oppii sanojen väliset suhteet muodostamalla dataan sisältyville sanoille vektoriesitykset.

Tutkielmassa selvitetään korpuksen koon vaikutusta sanojen samankaltaisuusmallien kouluttamisessa. Korpuksen koon vaikutusta tutkitaan vertaamalla mitattuja kosinin samankaltaisuuden etäisyyksiä sanojen vektoriesitysten välillä kahdessa vektoriavaruudessa, jotka ovat muodostettu hyödyntämällä erikokosia korpuksia. Työssä käytetään kahta korpusta, joista yksi sisältää vain viestejä Suomi24:n terveyskeskustelualueelta. Toinen korpus sisältää terveyskeskusteluviestien lisäksi viestejä muilta keskustelualueilta, jotka sisältävät terveysaiheisia sanoja.

Kosinin samankaltaisuutta arvioidaan hyödyntämällä käsitelistoja olennaisista terveysaiheisista sanoista. Kun datan määrää kasvatetaan lähes 30 prosenttia, huomataan, että kasvattamisella ei ole huomattavaa vaikutusta datan ominaispiirteisiin. Tulokset eivät osoittaneet selkeää yhteyttä korpuksen koon ja mitattujen terveysaiheisten sanojen vektoriesitysten kosinin samankaltaisuuksien välillä.

Avainsanat — Nyckelord — Keywords
sanojen vektoriesitys, word2vec, suomi24, tekstilouhinta, tiedonhaku

Säilytyspaikka — Förvaringsställe — Where deposited

Muita tietoja — Övriga uppgifter — Additional information

Contents

List of Figures

List of Tables

Introduction

Text mining or text retrieval is the process of extracting meaningful and actionable information from unstructured text data. The goal is to identify and locate the parts including information we are interested in. Text mining methods are solutions to how to solve the task of finding relevant items from text matching the specific information request (Lagus, 2000).

Today people produce enormous amounts of data when they interact with digital services in their everyday lives. Data produced by social media is one of the most interesting research topics as social media has emerged as a place to exchange information and experiences. Using social media produces data when people are publishing content on their own profile, but also browsing and other actions are logged by the services. Notably users are recording their own actions and feelings, when they are using social media in the form of written posts, messages or other kinds of updates. Hence social media offers a new kind of digital research area to study human behavior and interactions between other individuals.

One important resource in this digital research is Suomi24 online discussion data, which is a very extensive collection of messages posted to a popular Finnish speaking online forums since 2001. Discussion in the message boards of Suomi24 is usually very topic oriented compared to other social media platforms, where users mostly communicate using their own names. Anonymity provides privacy for open discussion as users can share their experiences and thoughts truthfully and be direct without worrying that their own messages would affect their social life offline.

For these reasons, Suomi24 provides a safe environment for people to discuss their health. People can address their own health concerns and seek peer support from others dealing with the same kind of health related issues. Most of all social media provides a new kind of point view of how people actually relate to medicine and their illnesses.

Using text mining methods, we can extract relevant parts of health discussion and analyze the texts to observe relations between these health related concepts, such as medication or symptoms. These collections of concepts are created utilizing data analysis, linguistic tools and limited human input to discover and recognize these relevant concepts from messages posted to the health section.

This thesis reviews how the training of word similarity models is affected by increasing corpus size using text retrieval methods. The effects of corpus size are examined by comparing the measured distance between word vectors representations in two different vector spaces. The focus is on how health related words, mainly words representing symptoms or drugs, are mapped in two different vector spaces. The location of the word vector representation reflects the semantic similarity between two words, which means that we can also examine how adding more training examples influences the location of defined word embeddings in vector space.

Machine learning methods are used to produce word vector representations for each word found in the training data. Using these word embeddings, words with similar vectors can be located and observed by computing the cosine similarity distance between these word vectors in a vector space. The training process of producing the word embeddings is repeated and a new model is trained with data including the health discussion data and the messages from other discussion areas containing relevant health domain words.

Chapter 1 introduces previous research about text mining social messages posted in Finnish discussion boards. Also characteristics of online health discussion are described and what relevant information can be extracted from these discussions using computational text retrieval methods. A novel method based on earlier research is introduced, which utilizes human input to classify word embeddings to produce concept vocabularies of domain specific relevant words.

Chapter 2 covers introduction to statistical language modeling and word vector representations. Suomi24 discussion dataset is described in more detail in Chapter 3. Results and findings are reported in Chapter 4.

The final Chapters 5 and 6 summarize the results and provide discussion on potential continuations of the research and how the evaluation process can be improved. Challenges and limitations encountered during the process of this thesis are considered and the solutions to solve these problems examined. Concluding with summarization of the results and how the results relate to the research questions of the thesis. In closing, the results are observed from the perspective of proposed future research.

Chapter 1

Background

Traditional medical research is focused on specific research questions such as the effectiveness and possible side effects or general safety regarding other unexpected health problems. Despite the extensive research conducted, drugs are not always perceived the same way intended by pharmacologists and other healthcare professionals. Direct access to internet-based medical advice and other types of digital health services has affected the traditional doctor-patient setting due to patients, or health consumers, being more informed (Autio et al., 2012; Hardey, 2001).

Since the early years of internet discussion, people have been discussing health, symptoms and drugs from their own perspective or sharing the experiences of someone close to them. Suomi24, a popular Finnish message board, has offered a platform for such discussion since it was launched back in 1998. Suomi24 discussions are openly available as a dataset, which includes every message posted since 2001[1]. The health discussion is one of the most common topics in Suomi24, which makes the health discussion alone a very valuable resource as a domain specific textual dataset of Finnish language(Lagus et al., 2016).

One important factor in this peer-to-peer online discussion is medication, which affects how health and illness are discussed (Ylisiurua, 2017). Earlier research project called Medicine Radar (also called Lääketutka in Finnish) researching the health discussion of Suomi24 was focused on exploring and visualizing the discussion itself and how drugs are perceived in the absence of healthcare professionals (Lagus et al., 2018).

This earlier research also focused on capturing words that represent the same concept, specifically drugs and symptoms and creating concept vocabularies using an augmented intelligence method for concept-oriented analysis. This method combines machine learning methods, such as Word2Vec (Mikolov, Chen, et al., 2013) to discover and capture concepts from the colloquial health discussions with human input, which was used to ultimately decide whether the captured word refers to

[1] In September 2020 data from 2001–2017 was available

the same specific concept.

These concepts are initially captured applying the distributed representations of words trained using only the health discussion as corpus. However the health discussion is not only limited to the health section of the forums. Using the concept vocabularies data can be supplemented by adding messages, which include health related discussion, but are posted outside the health section. A message posted outside health discussion is added to training corpus, if a message includes any of the word forms captured in the concept vocabularies.

These produced concept vocabularies are specific for the health domain and can also be used in evaluation of word embedding of the same domain. Using them in intrinsic evaluation of word embedding is not possible, since the concept vocabularies only include lexemes of the same lemma and possibly some prevalent misspelled word forms. However the captured concepts can be considered to be semantically related, when similarity is defined as co-hyponymy (Turney et al., 2010). This means that two words have any kind of semantic relation, i.e. if they share any kind of attributes, then two words are semantically related. Examples include words that are synonyms ("hospital" and "medical center") or words that are functionally or frequently associated ("paramedic" and "ambulance").

Motivation to generate such a collection is that as nothing equivalent exists in Finnish to our knowledge. Producing a comprehensive vocabulary manually would be time-consuming and as there are many fundamental issues that would require both medical and linguistic knowledge. For example, referring to the same medicine can depend on the context as medicines usually have multiple different marketing names or they are known by their colloquial nickname. Drug names can be easy to misspell and listing each possible misspelled form would take too much time. In addition, some can treat illegal substances as medication, which can be interpreted as personal medication even if it is not recommended by healthcare professionals. This novel method was presented originally in Lagus et al. (2018) provides a straight-forward solution to these issues by using computational methods, but retaining the final decision for human review.

Chapter 2

Theory

In this chapter, we focus on the theorectical aspects of how word vector representations are produced starting from introduction to statistical language modeling. The following sections focus on how word embeddings are produced using Word2Vec (Mikolov, Chen, et al., 2013) and how Word2Vec models can be used to predict contextually similar words. The most important parameters of Word2Vec training are explored from the perspective of word similarity tasks.

2.1 Statistical Language Modeling

Statistical language modeling is one way to process natural language input that allows computing probabilities for any sequence of character symbols. These character sequences can be anything from words to sentences or from paragraphs to text documents. In language modeling, the main interest is in probabilities of word sequences, and the goal is to assign a probability $P(w_{1:m})$ to any sequence of words $w_1, \ldots, w_m$. This can be achieved using the chain rule of probability and by conditioning each word in the sequence on its the preceding words. This is density estimation of the distribution of $P(w_{1:m})$ over word sequence $w_1, \ldots, w_m$ (Goldberg, 2017):

$$P(w_{1:m}) = P(w_1)P(w_2|w_1)\ldots P(w_m|w_{1:m-1}) = \prod_{i=1}^{m} P(w_i|w_1, \ldots, w_{i-1}) \tag{2.1}$$

Now each word is predicted conditioned on every preceding word. Using this formulation in computing is not demanding when predicting the next individual words, but modeling an entire sentence is harder, because this method would require computing all the possible sentences. Typically the solution is to use the Markov assumption, that states the future is independent of the past given the present. In other words, a kth order Markov-assumption assumes that the next word depends only on the last k words in the sequence:

$$P(w_{i+1}|w_{1:i}) \approx P(w_{i+1}|w_{i-k,\ldots,i}) \tag{2.2}$$

Now Equation 2.1 can be reformulated using the Markov assumption, where $k = i$:

$$P(w_{1:m}) \approx \prod_{i=1}^{m} P(w_i | w_{i-k,\dots,i-1}) \tag{2.3}$$

Even when conditioning on only the previous k words, the word order of the sequence has a substantial impact on results as this model does not take linguistic information into account, but only tries to match the given text pattern with the data. Also the conditional probabilities of words are not truly considered even under the Markov assumption, because possible dependencies beyond the window of k words are ignored.

2.2 N-gram Models

One of the simplest and also one of the most frequently used language models is the *N-gram model*. An N-gram is a sequence of adjacent words or letters of length N. More formally, N-gram models are based on the Markov assumption, where $k = N - 1$. For example, if we are observing the sentence *"Quick brown fox jumps over the lazy dog"* and we extract every *bigram* (an N-gram of size 2), which consists of two adjacent words, such as *"Quick brown"* or *"fox jumps"*. Respectively a *trigram* is a N-gram for $N = 3$ and it is a sequence consisting of three adjacent words.

The N-gram model itself is a probabilistic language model, which uses the statistical properties of N-grams to predict the next item in sequence, e.g. a sentence. The N-gram statistics are essentially based on co-occurence frequencies for words and are used for predicting the occurence of a word w_i from the occurence of the $N-1$ words $w_{i-(N-1)}, \dots, w_{i-1}$ preceding it. For example a bigram model predicts the conditional probability of the next word by conditioning on only one preceding word. Accordingly a trigram model approximates the same probability by using the conditional probability of past two words.

The conditional probability for words $w_i, \dots, w_m$ is computed the same way as in Equation 2.3, but now setting $k = N - 1$:

$$P(w_{1:m}) \approx \prod_{i=1}^{m} P(w_i | w_{i-(N-1),\dots,w_{i-1}}) \tag{2.4}$$

The maximum likelihood estimate of the probability of a word w_i in context $H = w_{i-(N-1)}, \dots, w_{i-1}$ can be acquired by counting the number of times w_i has appeared in the context H and normalizing it with the number of the N-grams including the context words (Jurafsky et al., 2008). First compute the count of the N-gram $C(\cdot)$ and normalize with the sum of all the N-grams that include the same words as in H:

$$\begin{aligned} P(w_i | w_{i-N+1}, \dots, w_{i-1}) &= \frac{C(w_{i-N+1}, \dots, w_{i-1}, w_i)}{C(w_{i-N+1}, \dots, w_{i-1})} \\ \iff P(w_i | H) &= \frac{C(H \times w_i)}{C(H)} \end{aligned} \tag{2.5}$$

As the probabilities for sentences that are not included in the training set would assigned a zero, we need to redistribute the probability mass from the most frequent occurences and redistribute it to events with otherwise zero probabilities. These methods are called smoothing techniques, of which the most common are presented in detail in Chen et al. (1999), Jurafsky et al. (2008).

2.3 Word Vector Space Representation

Early work on representing words in vector space were presented already in the 1990's, for example in Kohonen et al. (1996), Lagus (2000), Kohonen (2001). A new way to calculate the word vector representations that made the general approach widely popular was introduced by Bengio et al. (2003). Every word in the vocabulary is associated with a distributed word feature vector, which is a real-valued vector in $\mathbb{R}^n$. These vectors represent the different aspects and features of the word by assigning real valued numbers to the elements of the feature vector. Feature vectors that represent words as vectors can be also called simply *word vectors*. The term *word embedding* is also used to generally describe word vector representation and feature learning techniques that are based on representing words in a vector space as real valued points.

Word vector presentations are in principle similar to the RGB color model, which is one of the mostly used techniques to represent colors on digital screens. In the RGB color model, colors are composed by adding red, green and blue light together to create a new color. To put it simply, each of the three color channels (red, green, blue) are represented by a value, which resembles intensity of the light beam. Since each color channel is eight bits, each color channel has $2^8 = 256$ different possible values, which range from 0 to 255. This color model also allows labeling different colors and shades with more recognizable names to distinguish different colors more intuitively.

For example, a color considered to be *"Blue"* can be formulated vector $(0, 0, 255)$. Following this example, the color *"Cyan"* is represented by the vector $(0, 255, 255)$. Both *Blue* and *Cyan* are vectors of $\mathbb{R}^3$, basic arithmetical operations can be applied to these vectors. As we know that the color cyan can be created by mixing blue with green, we can also formulate it as an operation of adding vectors together to form a new vector:

$$\textit{blue} + \textit{green} = \textit{cyan} \iff (0, 0, 255) + (0, 255, 0) = (0, 255, 255)$$

Similarly these vectors can be multiplied or divided by real numbers to acquire a new color:

$$\textit{blue} \times \frac{1}{3} \iff (0, 0, 255) \times \frac{1}{3} = (0, 0, 85) = \textit{"navy blue"}$$

Of course this is just a simple example how to produce labels for different colors.

The same kind of approach applies to words, but instead of representing color channels, each element in word vectors represent a different dimension of the word's meaning. The dimensions of the vectors repsenting word's meaning can be observed as multidimensional continuous points in geometric space.

According to the *distributional hypothesis* words with similar contexts tend to have similar meanings. Contextually similar words are expected to have similar word vectors which implies that the points should be mapped to neighboring points in the same vector space.

Now we can apply linear algebra and other numerical operations to words using their representations in vector space, that we could not otherwise apply to textual data directly. For example, following the example in Mikolov, Sutskever, et al. (2013), if we have the words "Finland", "Helsinki", "Estonia" and "Tallinn", we could expect the following result when using the word vector representations $vec(\cdot)$ of these words:

$$vec(\text{“Helsinki”}) - vec(\text{“Finland”}) + vec(\text{“Estonia”}) = vec(\text{“Tallinn”})$$

This result suggests than when subtracting "Finland" from "Helsinki" and adding "Estonia" we would receive the word "Tallinn", because it would be the most similar vector with the vector we receive from this simple addition and subtraction operation.

The *context* for a word is usually taken to be the surrounding words in a given word sequence located in the training dataset. The distance between different words is taken into account when learning the weights based on how contextually close the words are. As mentioned above, elements in a word vectors represent different dimensions of the meaning of a word, but in more detail the elements are equivalent to the word's distributed weights across dimensions. The value of the weight defines how close the word in question is to that dimension's meaning.

In this word vector space we can measure the distance of two non-zero vectors using cosine similarity, which is a similarity measure defined to equal the cosine angle of the vectors projected in a multidimensional vector space:

$$\cos(\mathbf{A}, \mathbf{B}) = \frac{\mathbf{AB}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} \mathbf{A}_i \mathbf{B}_i}{\sqrt{\sum_{i=1}^{n} \mathbf{A}_i^2} \sqrt{\sum_{i=1}^{n} \mathbf{B}_i^2}} \tag{2.6}$$

In cosine similarity, larger values indicate higher similarity as the cosine angle becomes smaller.

2.4 Word2Vec

Word2Vec is a recent and popular embedding technique, which was proposed by Mikolov, Chen, et al. (2013). Word2vec is not a single model, but rather a group of models, which depend on the choice of model architecture and parameterization of that architecture. The two model architectures that can be used to train the model are the *Continuous Bag-of-Words* (CBOW) Model and the *Skip-gram Model*, and both of them are based on shallow neural networks, that consist of only of two layers.

Let $W = \{w_1, \ldots, w_T\}$ be a word sequence containing T words, represented as a set. In CBOW, the model predicts the target word w_t based on a given a set of surrounding context words $\{w_{t-n}, \ldots, w_{t-1}, w_{t+1}, \ldots, w_{t+n}\}$. The CBOW architecture is similar to the Feed-Forward Neural Network Language Models proposed by Bengio et al. (2003). Their language model used neural networks to learn the distribution word representations simultaneosly with the probability of words or phrases.

In the traditional language model, which predicts the next word based on the n preceding words, the word order matters. However in the CBOW model n words before and after the target word w_t are used. The preceding and following words of w_t are called the *surrounding context words* or the *context window*. The order of words in the context window does not matter in CBOW, because it uses the continuous word vector representations, which mean that computation of such vectors is commutative. On the contrary to CBOW predicting the next word given it surroudings, the Skip-gram model predicts the surrounding context words $\{w_{t-n}, \ldots, w_{t-1}, w_{t+1}, \ldots, w_{t+n}\}$ when given one target word w_t. The Skip-gram model will be covered in more detail in Section 2.5.

Choosing between these two architectures is essentially a task-specific decision, because the architectures have differences in two two important aspects: performance and accuracy, which should taken into consideration when dealing with large vocabularies and datasets. Whereas CBOW is considered faster, Skip-gram performs better learning uncommon and infrequent words according to studies by Tomáš Mikolov, the main author of Word2Vec (Mikolov, Chen, et al., 2013; Mikolov, Sutskever, et al., 2013). Other studies measuring the performance and comparing the architectures indicate better performace of Skip-gram compared to CBOW in some task settings (Lai et al., 2016; Th et al., 2015).

In this thesis we focus on Skip-gram, because Skip-gram vectors have shown better results in word similarity tasks compared to CBOW (Chiu et al., 2016).

Other important factors are the size of context windows c, the subsampling rate of frequent words and the number of dimensions used in the embedding process. The size of context window c can have an effect on accuracy, but it can also increase training time (Lison et al., 2017). In large vocabularies the most common words usually provide less useful information than rare words, because most common words would be often associated with many uninformative words such as articles

or conjunctions. However some of the problems concerning articles or conjunctions could be also solved applying preprocessing before making analysis from the data. As for the issue of word dimensionality, it has been shown that after certain point there is no notable gain in accuracy (Chiu et al., 2016).

In conclusion, the decision about the optimal configuration is extremely task specific and all of these aspects mentioned above should be taken into consideration.

2.5 Skip-Gram

The main object in the training of the Skip-gram model is to maximize the classification of one word in the sentence based on the other words in the sentence. Skip-gram tries to find other word representations that can be used to predict the surrounding context words given one word. The Skip-gram architecture is visualized in Figure 2.1.

The objective can be formalized to maximize the average log probability of a word being a context word given trainings words $w_1, w_2, w_3, \ldots, w_T$:

$$\frac{1}{T} \sum_{\substack{-c \leq j \leq c \\ j \neq 0}}^{C} \log \left(p(w_{t+j}|w_t) \right), \tag{2.7}$$

where w_t is the target word and$c$ is the size of context window (the number of context words).

Let V be the size of our vocabulary and N the dimensionality of word vectors. During training,the input word $\mathbf{v}_{w_I}$ is one-hot encoded into a vector $\mathbf{x} = \{x_1, \ldots, x_V\}$, which means that only one of the V elements in vector $\mathbf{x}$ is 1 and all other elements are zeroes.

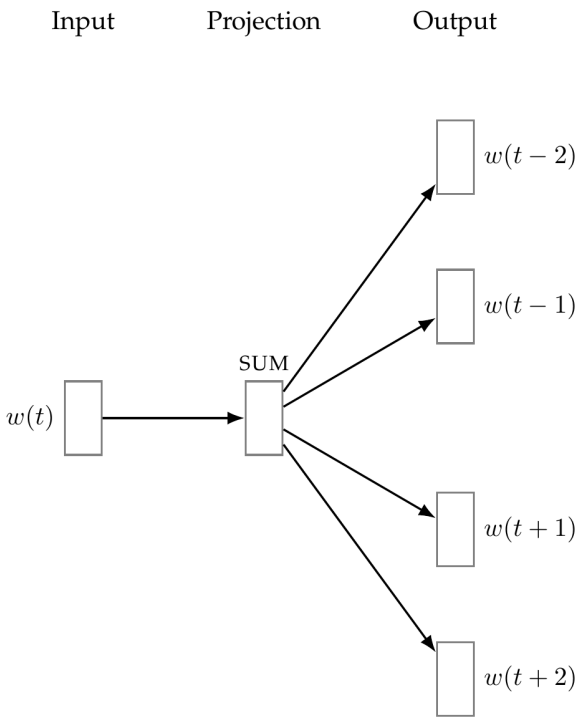

Figure 2.1: Skip-gram architecture

Learning of word embeddings in Word2Vec requires two weight matrices: the embedding matrix W of size $V \times N$ and the context matrix W' of size $N \times V$.[1] At first, the elements in both matrices W and W' are initialized with random values, but they are also updated during training. The update process is described in more detail in Section 2.6 and the model architecture with weight matrices is visualized in Figure 2.2.

Each row of the embedding matrix W is an N dimensional vector representation $\mathbf{v}_w$ for one vocabulary word w. Likewise every input word w_I is also in the vocabulary, so the location of the non-zero element x_k in vector $\mathbf{x}$ corresponds to the index of the input word's $\mathbf{v}_{w_I}$ location in the vocabulary.

Now because vector $\mathbf{x}$ is binary and we know that the only non-zero element is the kth element, we can simply take only the kth row from W instead of multiplying the whole matrix W with vector $\mathbf{x}$. This N dimensional vector projection of input word w_I will be the hidden layer h:

$$h = W^T \cdot x = W^T_{(k,.)} := v^T_{w_I} \tag{2.8}$$

Next the vector projection $h \in \mathbb{R}^N$ will be multiplied with ${\mathbf{v}'_{w_j}}^T$, the jth column of context weight matrix W', to get the score $u_{c,j}$, which measures how close input

[1] Regardless of notation, the context weight matrix W' is not a transpose matrix of W, and it's consider to be independent of W.

words are to the context window. Even though the jth column of matrix W' is the vector representation of the meaning of the word w_j in the vocabulary, these weight are not outside the training process.

We receive the net input of every unit j on every panel c of the output layer. We know that each output panel c uses the same weight matrix, which means that scores u_j for each word w_j are the same for every panel c, so

$$u_{c,j} = u_j = {v'_{w_j}}^T \cdot h \text{ , for } c = 1, 2, 3 \ldots, C. \tag{2.9}$$

At the output layer, to obtain the posterior distribution of words from the scores $u_{c,j}$, we need Softmax, a log-linear classification model, to normalizes the values to probabilities:

$$\text{Softmax}(x_i) = \frac{\exp x_i}{\sum_j \exp x_j} \tag{2.10}$$

Now $p(w_{t+j}|w_t)$ (2.5) is the conditional probability of observing a context word (outside word) w_O given input word (target word) w_I and can be reformulated using Softmax:

$$\begin{aligned} p(w_{c,j} = w_{O,c}|w_I) &= y_j \\ &= \text{Softmax}(u_{c,j}) \\ &= \frac{\exp\left(u_{c,j}\right)}{\sum_{j'=1}^{V} \exp\left(u_{j'}\right)} \\ &= \frac{\exp\left({v'_{w_O}}^T v_{w_I}\right)}{\sum_{i=1}^{V} \exp\left({v'_{w_i}}^T v_{w_I}\right)}, \end{aligned} \tag{2.11}$$

where v_w and v'_w are the input and output vector representations and V the size of vocabulary.

The dot product in the numerator, ${v'_{w_O}}^T v_{w_I}$, measures the similary between vector representations of the outside word W_O and the target word w_I. Values produced by this dot product express similarity between these vectors, as when both vectors have large values in the same dimensions, the resulting value will be high, see for example Jurafsky et al. (2008).

On the other hand, the dot product in the denominator ${v'_w}^T v_{w_I}$ is computational expensive, because it requires computing dot products between target word w_I and all other words in vocabulary. Computing these dot products make the training impractical and time consuming with large vocabularies. Mikolov et al proposed two new training algorithms in Mikolov, Sutskever, et al. (2013): hierarchical softmax and negative sampling, which both optimize the computation of the updated output vectors. These two training algorithms are applicable to both Skip-gram and CBOW, but in this thesis we will be introducing them only using Skip-gram, because we are focusing on the similarity between target words and model generated context words.

2.6 Updating Weights

To find the optimal values for both weight matrices W and W', prediction error of words must be minimized during training. A loss function evaluates how well words are modelled from the data. The objective is to maximize the probability of output words $w_{O,1}, \ldots, w_{O,C}$ given the input word w_I:

$$\begin{aligned}
\max p(w_O|w_I) &= \max y_{j^*} \\
&= \max \log y_{j^*} \\
&= \log \left(p(w_{O,1}, \ldots, w_{O,C}|w_I)\right. \\
&= \log \prod_{c=1}^{C} \frac{\exp(u_{c,j_c^*})}{\sum_{j'=1}^{V} \exp\left(u_{j'}\right)} \\
&= \sum_{c=1}^{C} u_{c,j^*} + \sum_{c=1}^{C} \log \sum_{j'=1}^{V} \exp(u_{c,j}) \\
&= \sum_{c=1}^{C} u_{j_c^*} + C \cdot \log \sum_{j'=1}^{V} \exp(u_{j'}) := -E,
\end{aligned} \tag{2.12}$$

where j^* is the index of the cth output word in the output layer and E is the loss function, which is being minimized.

The loss function of Skip-gram E is optimized in using Stochastic Gradient Descent (SGD), which uses the error gradients of the loss function to learn optimal values of a weight matrix. In addition, because computation with all vocabulary words during every iteration is computationally expensive, with SGD the weights can be updated one training word w_t at a time. Thus the values of the hidden units h are updated at the same time.

2.6.1 Update Equation for Context Matrix

The error derivative of output layer's input is obtained by taking the derivative of E with respect to the net input of panels $u_{c,j}$to the prediction error of the output layer $e_{c,j}$ (Rong, 2016):

$$\frac{\partial E}{\partial u_{c,j}} = y_{c,j} - t_{c,j} := e_{c,j}, \tag{2.13}$$

where $t_{c,j}$ is a indicator function $\mathbb{1}_{c,j}\{j = j^*\}$, where $t_j = 1$ only when the jth unit is the actual output word, otherwise $t_j = 0$.

The second required error derivative is the derivative of the loss function E with respect to the weight matrix of the output layer W':

$$\frac{\partial E}{\partial w'_{ij}} = \sum_{c=1}^{C} \frac{\partial E}{\partial u_{c,j}} \cdot \frac{\partial u_{c,j}}{\partial w'_{ij}} = EI_j \cdot h_i, \tag{2.14}$$

where $EI_j = \sum_{c=1}^{C} e_{c,j}$ is the sum of prediction errors of every context word c and h_i is expanded from Equation (2.8):

$$h_i = \sum_{k=1}^{V} x_k w_{ki} \tag{2.15}$$

Let $\eta > 0$ be the learning rate, then the update equation for weights ${w'}_{ij}$ in context matrix W' is:

$$\begin{aligned} {w'}_{ij}^{(new)} &= {w'}_{ij}^{(old)} - \eta \cdot \nabla E \\ &= {w'}_{ij}^{(old)} - \eta \cdot EI_j \cdot h_i \end{aligned} \tag{2.16}$$

This update equation requires checking the probability $y_{c,j}$ for every "vocabulary" word and comparing it with its estimate $t_{c,j}$. The values of w'_{ij} are adjusted by subtracting or adding proportion of h_i determined by prediction error $e_{c,j}$ on the basis of how well $y_{c,j}$ is estimated.

2.6.2 Update Equation for Embedding Matrix

Following Equation (2.16), the weight in embedding matrix W are updated in similar way by taking the derivative E with respect to the output of the hidden layer h_i (Rong, 2016) :

$$\begin{aligned} \frac{\partial E}{\partial h_i} &= \sum_{j=1}^{V} \frac{\partial E}{\partial u_j} \cdot \frac{\partial u_j}{\partial h_i} \\ &= \sum_{j=1}^{V} \sum_{c=1}^{C} e_{c,j} \cdot w'_{i,j} \\ &= \sum_{j=1}^{V} EI_j \cdot w'_{i,j} := EH_i \end{aligned} \tag{2.17}$$

Now we obtain the sum of output vectors of every vocabulary word weighted by their prediction error fo $EH = \{EH_1, \dots EH_N\}$

The second derivative is computed with respect to every weight in W:

$$\begin{aligned} \frac{\partial E}{\partial w_{ki}} &= \sum_{j=1}^{V} \frac{\partial E}{\partial h_i} \cdot \frac{\partial h_i}{\partial w_{ki}} \\ &= \underbrace{\sum_{j=1}^{V} \sum_{c=1}^{C} e_{c,j} \cdot w'_{i,j}}_{=EH_i} \cdot x_k \\ &= EH_i \cdot x_k \end{aligned} \tag{2.18}$$

Thus the update equation for embedding matrix W is:

$$\begin{aligned} w_{ij}^{(new)} &= w_{ij}^{(old)} - \nabla E \\ &= w_{ij}^{(old)} - \eta \cdot EH^T, \end{aligned} \tag{2.19}$$

where w_{ij} is the only non-zero row in W, which will be updated while other rows in W are not updated this instance.

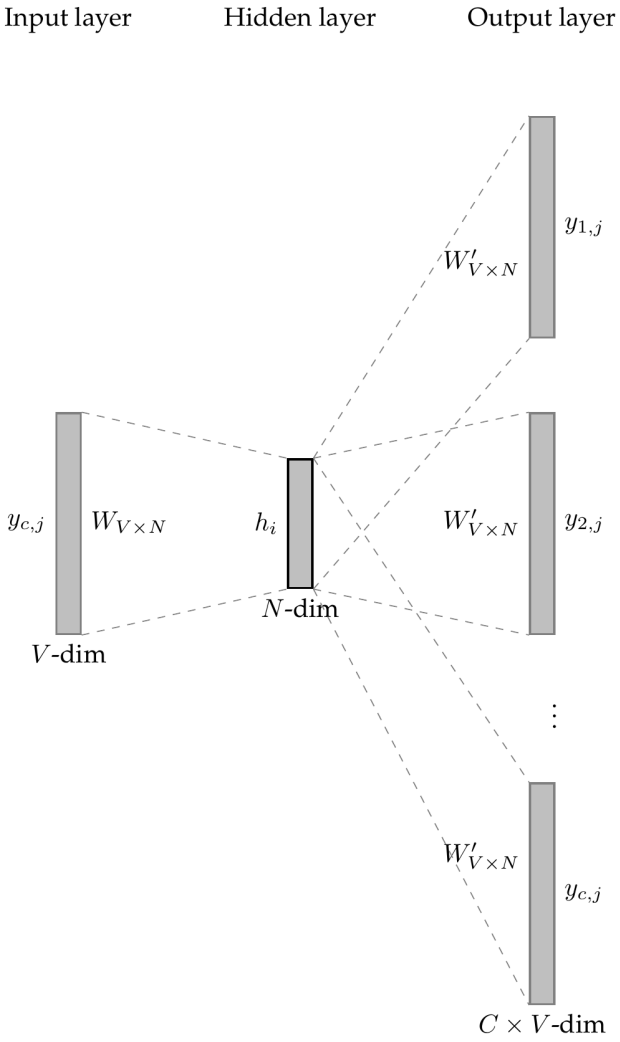

Figure 2.2: Skip-gram architecture with weight matrices

2.7 Hierarchical Softmax

Learning the output vectors is very expensive, because updating $\mathbf{v}'_w$ for every vocabulary word w_j we need to compute net input u_j, probability prediction $y_{c,j}$, prediction error EI_j. Particularly when vocabulary size is large, the number of required iterations increases, which affects the training time considerably. In order to reduce training time and to optimize computational efficiency, Word2Vec can be trained using two different training algorithms: negative sampling or hierarchical softmax.

Negative sampling is more straightforward solution based on noise contrastive estimation, where the weights are updated using sample negative examples. The output word is considered as positive sample and the log-likelihood ((2.12) is minimized using negative sampling. Main benefit of using negative sampling is to learn accurate representation for common words in the corpus (Mikolov, Sutskever, et al., 2013).

In this section we introduce hierarchical softmax (Mnih et al., 2008; Morin et al., 2005), where $|V|$ dimensional output softmax layer is replaced with a binary Huffman tree representation, where the leaves of the tree represent the vocabulary words. A unique path from root to each leaf exist and the paths are used to estimate the probability of the word, which are represented by the leafs of the Huffman tree (Figure 2.3). Similar paths in the tree are assigned with similar probabilities, which means that rare words will inherit their parent vector representations in the tree. Thus word vector representations of infrequent words are influenced by the more frequent words in the same corpus, which also rectifies the preference of updating weights with respect to the most common words in the corpus.

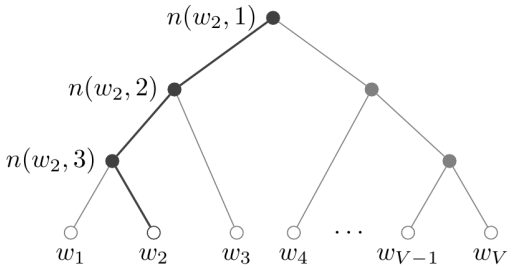

Figure 2.3: An example binary Huffman tree for the hierarchical softmax model.

Now each of the $V - 1$ inner units of the tree have an output vector $v'_{n(w,j)}$, which replace the output vector representations for words. The probability of a word being the output word w_O is:

$$p(w = w_O) = \prod_{j=1}^{L(w)-1} \sigma\left([\![n(w, j+1) = \mathrm{ch}(n(w, j))]\!] \cdot {v'_{n(w,j)}}^T h\right), \tag{2.20}$$

where ch(n) is the left child of unit n, $v'_{n(w,j)}$ the vector presentation of the inner unit $n(w, j)$, $h = v_{w_I}$ is the output value of the hidden layer and $[\![x]\!]$ is an indicator function:

$$[\![x]\!] = \begin{cases} 1 & \text{if } x \text{ is true,} \\ -1 & \text{otherwise} \end{cases}$$

and $\sigma(x)$ is

$$\sigma(x) = \frac{1}{1 + \exp(-x)}$$

We can replace $p(w_{c,j} = w_{O,c}|w_I)$ Equation (2.11) with (2.20) and use it to maximize the probability of a word being an output word w_O, which means that hierarchical softmax is a multinomial distribution among all words. Basically we try to predict should we follow the path to the left of the right during tree traversal.

Complexity is now reduced to $\mathcal{O}(\log(|V|))$ from $\mathcal{O}(|V|)$, making the faster training time the main improvement over regular softmax. Because of taking less time to train, now the training algorithm can observe even the rarest words and adjust the weights more often compared to a model trained with regular softmax trained in the same time (Chen et al., 2018).

Chapter 3

Data

In this chapter, Suomi24 is introduced as a social media and the discussion in the forums are described. The following subsections focus on preprocessing the data from computational approach, before any natural language processing methods can be applied.

3.1 Suomi24

Suomi24 is one of the Finland's largest social networking services, which has been operating since 1998. Over the recent 20 years, the discussion forums have established its position as the most popular and commonly recognized service of Suomi24. Today Suomi24 has around 1.9 million monthly visitors, which is roughly 41% of total internet-users in Finland[1]. In 2017, on average 10 600 new messages were posted each day of the year.

Key characteristic for taking part in discussion is that the discussion forums do not require any registration, but users have to select a nickname. These nicknames are not unique and they can be used interchangeably by different people. If a user decides to register, then the selected nickname cannot be chosen by unregistered users. Only under 10% of users are registered (Lagus et al., 2016). Privacy provided anonymity allows discussion on Suomi24 is described to be more focused on the discussion topic rather than creating new or maintaining current social connections of the user. Anonymity allows users to write about difficult and intimate topics truthfully, but on the other hand it lowers the bar of misbehaving and trolling of other users.

All the messages from Suomi24 discussion boards are openly available for research purposes from the Language Bank of Finland. Data is provided by initial cooperation with Aller Media, the former owner of Suomi24 until December 2019. The promised update cycle was set to every 6 months and the newest messages from

[1] FIAM December 2019

the message boards posted during this time period would be added to data. At the moment, the dataset includes years 2001–2017.

The dataset does not contain any background information about the users, which means that identifying users demographically is not possible. Messages posted by unregistered users cannot be traced or linked with any individual users and one users can therefore use several different nicknames when participating in the discussion.

The dataset is considered one of a kind due its time span, coverage of various topics and the remarkable number of sent messages. It consists mostly everyday conversation in colloquial Finnish in a written text format. The language used in the forums represents rather spoken Finnish than written Finnish, which means that it records interaction of users digitally in the same way that these discussions would take place person to person. Discussion forums also expanded the reach beyond the scope of traditional change of thoughts taking place offline, since the emergence of social media enable more people to be engaged with the discussion. Having the access to messages posted as early as 2001 is itself also a valuable resource to be explored in various fields of research, i.e. linguistics or social sciences.

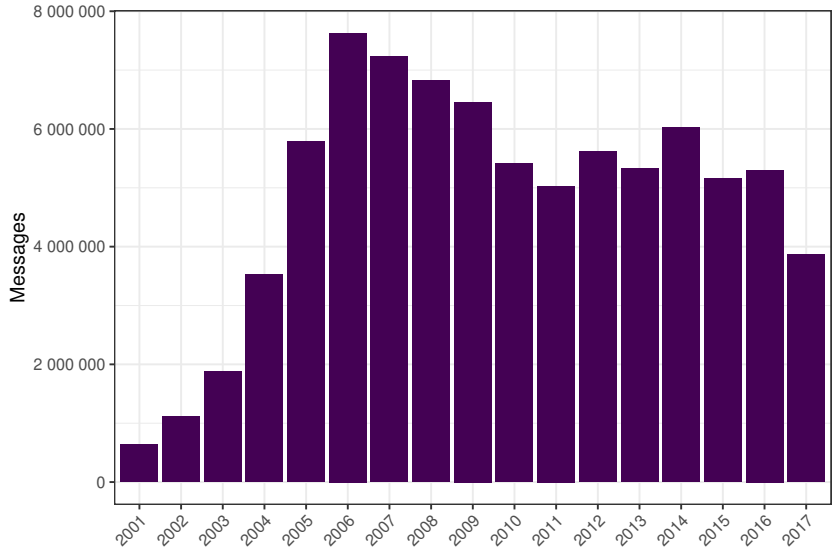

Figure 3.1: Distribution of messages posted to Suomi24 by year

3.2 Health Discussion

This thesis focuses on messages posted to the health discussion section of Suomi24, which one of the most popular topics in the forums overall (Lagus et al., 2016). The possibility to write anonymously and the fact that users cannot be recognized in

the forums allows users to share information that they would not normally want to share publicly, specifically regarding health issues of their own for social support (Lagus et al., 2015). Additionally users are interested in information on specific treatments or drugs and possible adverse effects. Health related forums provide a suitable platform for such peer to peer discussion of stigmatic health concerns, that are otherwise not easy to be openly addressed.

Medical internet discussion is partially a new way to address your own concern and discuss them with others. Important element in online health discussion is that is based on peer to peer discussion, where all participants are treated as equal. In the conventional setting, such as doctor's appointment, the other side of conversation is a healthcare professional and has medical expertise. In particular, the discussion is mostly focused on medication, which is central to how people discuss health and illnesses (Ylisiurua, 2017). Talking without a presence of professional users can explore openly their personal emotions and relationships with medication. Medication is often also criticized for being either inadequate or too excessive and causing either medical malpractice or overdiagnosing, which are both relevant concerns of the healthcare system.

Online discussion does not replace the medical advice given by professionals, but users having access to vast amount of information online including medical information, has also affected the general doctor-patient -relationship (Autio et al., 2012). Having this kind of access has made patients more aware of their own health by searching information online more independently. The open access to medical information has also enabled false or possibly dangerous information to spread more easily.

3.3 Medicine Radar

Lagus et al. (2018) developed a tool called Medicine Radar[2] to explore Suomi24 health discussion for people with no further technical skills. Medicine radar generally depicts the main points and topics of discussion by focusing on common concerns about how different drugs are perceived and how they affect the everyday lives of individuals.

Many parts of this thesis are based on the previous work done in the Medicine Radar project. Medicine Radar included two interesting details from text mining perspective:

1. **Word2Vec model**, which was trained with health discussion messages and used to assess similarity of two words and to look for other health related concepts.
2. **Concept vocabularies of symptoms and drugs**, which are produced with methods described in Lagus et al. (2018).

[2]Also known as Lääketutka in Finnish. Available at laaketutka.fi

The initial idea was to utilize the previous work done in Medicine Radar by using the same Word2Vec model for comparison with other models trained with different sized random samples from the same data. However, after trying to replicate the setting and use the same data to repeat the training of a Word2Vec model, this setting turned out practically impossible due to various reasons regarding documentation and the updates in the data structure made after finishing the original project.

The original idea, when starting this thesis work, was to evaluate the effects of corpus size by training models with less data. Instead the focus was set to applying the same method for generating domain vocabularies, which was introduced and used in Medicine Radar project, to expand the data with messages outside the health discussion area. The idea is to identify messages including health related words, but which are posted to other discussion areas. This is in a way the opposite of the original idea, where the focus was to simply research the effect of corpus size by gradually increasing the sample size of messages chosen randomly from orginal data and comparing the results with the model trained with all avalaible data. The focus is now on how the word vector representations are affected when messages from other domains are included also in the data. The level of noise in the data can increase for this reason as the discussion context can be very different from the medicinal context that we are mostly interested in. The evaluation process is describing in more detail in Chapter 4.

3.4 Preprocessing

The Suomi24 discussion data is available to download in verticalized text format (VRT), which is a token-oriented columnar text format. Each line includes a single word and its annotation attributes, such as lemma or part of speech, which are separated by tabs. The structure of the text follows XML-formatting style, where structural attributes are represented by tags, which can include also XML-style attributes. Syntactic annotations in the data have been created using Turku Dependency Treebank[3].

VRT allows spaces in structure tags, whereas regular XML, where attributes are name tokens, does not support spaces in attributes. Implementations of loading and importing VRT files in to statistical computing environments, such as R or Python, did not exist to our knowledge or they were not directly applicable. Also the VRT version of Suomi24 data has been updated after completing work on Medicine Radar. This resulted in multiple issues with the data that was originally used in the Medicine Radar project, such as discrepancies in dependency annotations. Reproducing exactly the same data was considered too time-consuming and despite the efforts, there would be no guarantee that the we would receive similar results as in Lagus et al. (2018). In addition to possible reproducibility problems,

[3] `https://bionlp.utu.fi/fintreebank.html`

the newer VRT version 1.1 offers improvements over the older version, such as providing each calendar year as single file and empty and dummy messages are removed. Unlike in Medicine Radar, the year 2017 is included in the data. The

```
<text comment_id="0" date="2001-01-01" datetime="2001-01-01 20:11:00" author="Mari" parent_comment_id="0"
quoted_comment_id="0" author_logged_in="n" nick_type="anonymous" thread_id="44725" time="20:11:00"
title="Kuoleminen unessa." topic_nums="3289,2692,2795,4" msg_type="thread_start" topic_name_leaf="Uni ja
unihairiöt" topic_name_top="Terveys" topic_names="Terveys &gt; Henkinen hyvinvointi ja mielenterveys &gt;
Uni ja unihäiriöt &gt; Uni ja unihairiöt" topic_names_set="|Henkinen hyvinvointi ja mielenterveys|Terveys|
Uni ja unihäiriöt|Uni ja unihairiöt|" topic_nums_set="|2692|2795|3289|4|" topic_adultonly="n"
datefrom="20010101" dateto="20010101" timefrom="201100" timeto="201100" id="44725:0" author_v1="Mari"
author_name_type="user_nickname" author_nick_registered="n" author_signed_status="0"
thread_start_datetime="2001-01-01 20:11:00" filename_vrt="s24_2001_01.vrt" parent_datetime=""
datetime_approximated="n" empty="n" filename_orig="threads2003a.vrt" origfile_textnum="38904">
<paragraph id="711" type="title">
<sentence id="1726">
Kuoleminen      1       kuolla  kuolla  N       NUM_Sg|CASE_Nom|DRV_Der_minen|CASECHANGE_Up     0
ROOT    _       1       |kuolla..nn.1|
unessa  2       uni     uni     N       NUM_Sg|CASE_Ine 1       nommod  SpaceAfter=No   2       |uni..nn.1|
.       3       .       .       Punct   _       1       punct   SpacesAfter=\n\n        3       |...xx.1|
</sentence>
</paragraph>
<paragraph id="712" type="body">
<sentence id="1727">
Näin    1       näin    näin    Adv     CASECHANGE_Up   5       advmod  _       1       |näin..ab.1|
noin    2       tuo     tuo     Pron    SUBCAT_Dem|NUM_Pl|CASE_Ins      4       quantmod        _
2       |tuo..pn.1|
vuosi   3       vuo     vuo     N       NUM_Sg|CASE_Gen|POSS_PxSg2      5       nommod  _       3       |
vuo..nn.1|
sitten  4       sitten  sitten  Adv     _       3       adpos   _       4       |sitten..ab.1|
unta    5       uni     uni     N       NUM_Sg|CASE_Par 0       ROOT    SpaceAfter=No   5       |uni..nn.1|
,       6       ,       ,       Punct   _       8       punct   _       6       |,..xx.1|
jonka   7       joka    joka    Pron    SUBCAT_Rel|NUM_Sg|CASE_Gen      8       rel     _       7       |
joka..pn.1|
muistan 8       muistaa muistaa V       PRS_Sg1|VOICE_Act|TENSE_Prs|MOOD_Ind    5       rcmod   _
8       |muistaa..vb.1|
hyvin   9       hyvin   hyvin   Adv     _       10      advmod  _       9       |hyvin..ab.1|
vieläkin        10      vieläkin        vieläkin        Adv     _       8       advmod  SpaceAfter=No
10      |vieläkin..ab.1|
.       11      .       .       Punct   _       5       punct   SpacesAfter=\n  11      |...xx.1|
</sentence>
<sentence id="1728">
Ajoin   1       ajaa    ajaa    V       PRS_Sg1|VOICE_Act|TENSE_Prt|MOOD_Ind|CASECHANGE_Up      0
ROOT    _       1       |ajaa..vb.1|
autoa   2       auto    auto    N       NUM_Sg|CASE_Par 1       dobj    SpaceAfter=No   2       |auto..nn.
1|
.       3       .       .       Punct   _       1       punct   _       3       |...xx.1|
</sentence>
```

Figure 3.2: Example of one Suomi24 message saved in VRT file format

dataset including years 2001–2017 consist of 17 files totaling 409.2 gigabytes (GB) in size. File sizes range from 3.4GB to 39.3 GB. These files are processed as text files and read line by line into memory until a line with `</text>` closing tag is read. After this the read lines are combined in to single string of characters and spaces from XML attributes are removed in order to process the strings as regular XML strings. Processing messages as XML strings allows straight-forward extraction of the message attributes, in this case topic in which the post was originally posted in. This information is used to acquire the distribution of messages by topic, which can be examined in Figure 3.2. At first, the string containing metadata and the message itself is discarded if it was not posted in the health discussion section. After this each paragraph from the message is processed so that each word form is extracted from each line line and to compose a string including the original message as it was written originally. In addition, from each captured line we extract also the lemma

Table 3.1: Discussion topic distribution of Suomi24 and the expanded data

		Entire Suomi24 data		Expanded data	
Topic	English	No. Messages	%	No. Messages	%
Ajanviete	*Pastime*	1 865 077	2.25 %	8 684	0.20 %
Ajoneuvot ja liikenne	*Vehicles and Traffic*	5 597 208	6.76 %	19 821	0.46 %
Harrastukset	*Hobbies*	2 837 355	3.42 %	6 181	0.14 %
Koti ja rakentaminen	*Home and Building*	2 205 838	2.66 %	6 872	0.16 %
Lemmikit	*Pets*	1 886 413	2.28 %	44 862	1.05 %
Matkailu	*Traveling*	1 653 976	2.00 %	15 649	0.37 %
Muoti ja kauneus	*Fashion and Beauty*	573 714	0.69 %	6 907	0.16 %
Nuoret	*Youth*	1 321 794	1.60 %	7 773	0.18 %
Paikkakunnat	*Local*	8 041 052	9.70 %	32 369	0.76 %
Perhe	*Family*	2 318 156	2.80 %	54 687	1.28 %
Ruoka ja juoma	*Food and Drink*	375 875	0.45 %	3 287	0.08 %
Ryhmät	*Groups*	2 933 842	3.54 %	24 156	0.57 %
Seksi	*Sex*	1 462 593	1.77 %	11 686	0.27 %
Suhteet	*Relationships*	10 036 730	12.11 %	34 694	0.81 %
Talous	*Economy*	1 276 627	1.54 %	4 427	0.10 %
Terveys	*Health*	3 788 232	4.57 %	3 788 232	88.64 %
Tiede ja teknologia	*Science and Technology*	4 514 735	5.45 %	19 554	0.46 %
Työ ja opiskelu	*Work and Studies*	1 760 080	2.12 %	8 550	0.20 %
Urheilu ja kuntoilu	*Sport and Exercise*	1 421 601	1.72 %	10 943	0.26 %
Viihde ja kulttuuri	*Entertainment and Culture*	4 746 767	5.73 %	30 003	0.70 %
Yhteiskunta	*Society*	22 240 943	26.84 %	134 437	3.15 %
Total		**82 858 608**	**-**	**4 273 774**	**-**

of each word. Each message is therefore available in the original spelling, but also in lemmatized format. Both messages are lowercased to avoid any problems interpreting words caused by different letter cases of the same word. Punctuation marks are padded with spaces on both sides. Identifiable information, such as user names, email addresses of registered users are removed. Every unnecessary formatting is removed by replacing them with either spaces or empty characters. Stop words are also removed before using the data in Word2Vec training.

3.5 Concept Vocabularies

To extract relevant concepts, identifying different forms of the words representing these concepts is a common problem in information retrieval. For health discussion we focus mainly on capturing medicines and symptoms, which both represent the most relevant concepts in the health domain. To capture the correct words we have to not only consider whether a word represents a medicine or a symptom, but do two words refer to the same medicine or symptom.

As explained in Lagus et al. (2018), such vocabulary does not exist in Finnish and creating something from scratch would require a lot of manual effort. In the method introduced in the Medicine Radar vocabularies were constructed by user selecting the initial input words. Word2Vec model trained with messages from the health discussion suggests other similar words using cosine similarity (2.6) to find the closest vector representation of another word. Word2Vec models are able to find semantically and syntactically similar words in a large corpus, but in this process the main focus is in semantics. User then can decide whether or not the suggested word represents the input words or not.

After the suggestions start to deteriorate from the seed word's meaning, user can stop and move to the next seed word. All the accepted words are also used as new seeds for future searches. The vocabularies can be supplemented manually by adding words using i.e. regular expressions or by parsing a list of medicine names available online. Finally the drug list is stemmed and the symptom list is both stemmed and lemmatized.

We use the exact same keyword lists produced in Lagus et al. (2018), which are openly accessible[4]. Since we naturally could not exactly replicate the augmented AI (humanand articial inteligence) interactive process of seed word creation, we decided to instead use the exact same keyword lists as were already produced by this process in Lagus et al. (2018). However, the Word2Vec model that we use is naturally not exactbly the same, so direct comparison in vector spaces is not possible. The Word2Vec model used in the process described above was trained using the health discussion messages from 2001–2016 and is therefore not the same model that is used in the evaluation section of this thesis. Figure 3.3 presents the

[4] `https://github.com/futurice/laaketutka-prereqs`

occurrences of the most common concept vocabulary words in the health discussion data and in the expanded data. Also the word occurrences from Medicine Radar project are included in comparison.

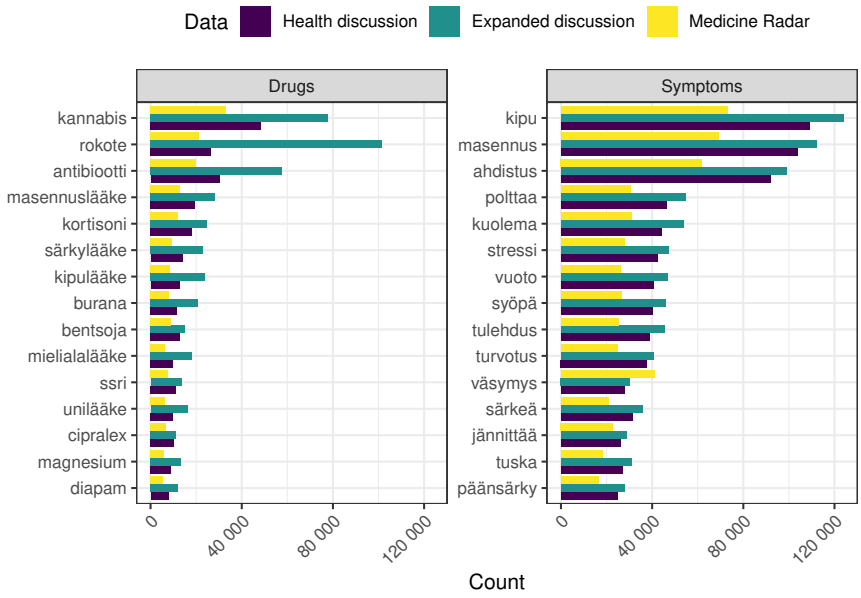

Figure 3.3: Most common drug and symptom words

These two lists can be considered as a collection of relevant health concepts in Finnish language. Generation process involves human decisions around the semantics of the words, but word vector representation provide a tool to explore the semantic relations. Currently, there are no simple and straightforward method or applicable language resources in Finnish to be used to solve problems around interpreting language using only computational data mining methods. Most common problems are interpreting ambiguous words especially if they share a stem or lemma with another word meaning a totally different thing. Also Word2Vec models can map two words that are related to each other, but they mean different things, such as a two drug classes can be related to each other, but they refer to different types of medication.

Chapter 4

Results

4.1 Models and Training Data

Word2Vec model parameters and training datasets are summarized and described in Table 4.1. Model 1 was trained using only messages from health discussion and Model 2 was trained using the expanded data including the same messages as Model 1, but also messages from other discussion areas. Context window size is set 5 and also words that appear less than 5 times in the data are not used in training. Both models were trained using the lemmatized forms of words. Vocabulary size $|V|$ is 28.6 % larger when using expanded data. The number of messages grew by 12.9 %, but there is no noticeable change in the number of words in an average message. Based on these numeric statistics, expanding the data does not affect the characteristics of captured messages in terms of number of words included to trainings.

Comparing two word vector similarity models is not very straightforward, since there is no ground truth model that captures every word accurately. Model performance can not be evaluated using metrics like precision, recall or accuracy, because text retrieval is a human-centered process, which makes performance evaluation difficulties. It would require classification by human experts, who can identify and classify items as relevant or non relevant to rigorously evaluate performance of models. In this thesis, using such experts to classify drug or symptoms words was not possible. Ideally a list of word semantic relations, where items are representing the key concepts in a certain domain would be a suitable solution. Word2Vec models could be evaluated using intrinsic evaluation by comparing the expected results with the result produced by the model. In our knowledge this kind of list did not exist in Finnish language and producing such a list would require expertise in pharmacology and linguistics, which means that we had to utilize other existing resources.

Both models produce 100 dimensional vectors for each word in their vocabulary, but we cannot make any conclusions based on the values, i.e. compare the value in

Table 4.1: Training data and Model architectures and parameters

	Model 1	Model 2
Training data		
Years	2001-2017	2001-2017
Threads	549 484	848 076
Messages	3 761 198	4 246 740
Avg. length of message (std)	43.26 (61.45)	48.78 (104.64)
Min	0 words	0 words
25%	12 words	13 words
Median	27 words	28 words
75%	54 words	57 words
Max	10 538 words	11 862 words
Model architecture	Skip-Gram	Skip-Gram
Word2Vec model		
Training algorithm	Hierarchical Softmax	Hierarchical Softmax
Window size	5	5
Dimensions	100	100
Minimum count	5	5
Resulting vocabulary size	*638 821*	*821 374*

the same dimension in two different vector spaces. However we can visualize the vector as a heatmap. The 15 most frequent drugs and symptoms are visualized in Figure 4.3. English translations of these words are available in Table 4.2.

In this figure, the color represents the value Word2Vec has mapped for each word embedding. We can notice that the majority of dimensions include similar values. Most prominent remark is that the word “magnesium” has the most deviant values in some dimensions in both models. This is most probably due to the fact that magnesium can be interpreted as the element used in various other purposes, but also in medicinal use. From symptom words, we can observe that word “vuoto” (leakage, drain) also deviates from other words, because it has a different meaning in other contexts. Unexpectedly, its word embedding seems more similar with other symptom words in Model 2 than in Model 1. In Figure 4.2 one notable feature that differentiates the Expanded dataset from both Medicine radar original data as well as from including only the health discussions is the rising of rokote (inoculation) as the most frequent word from the second place.

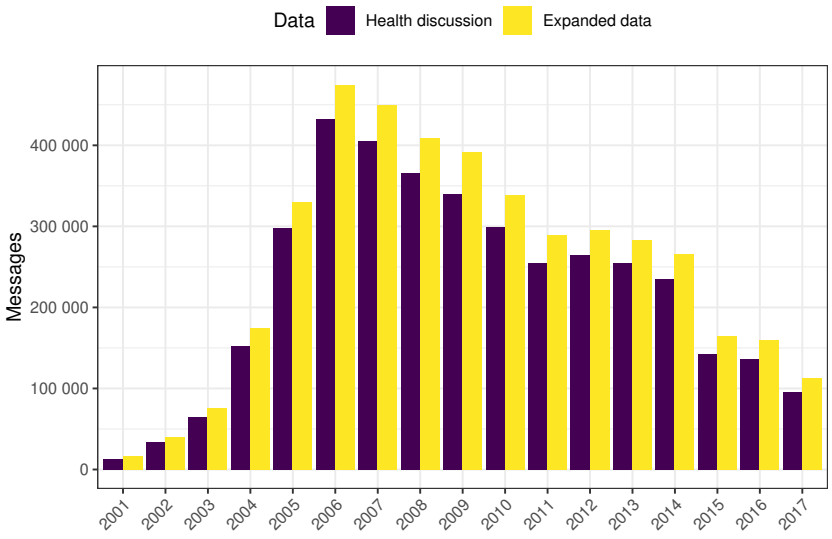

Figure 4.1: Distribution of new messages included in training data by year

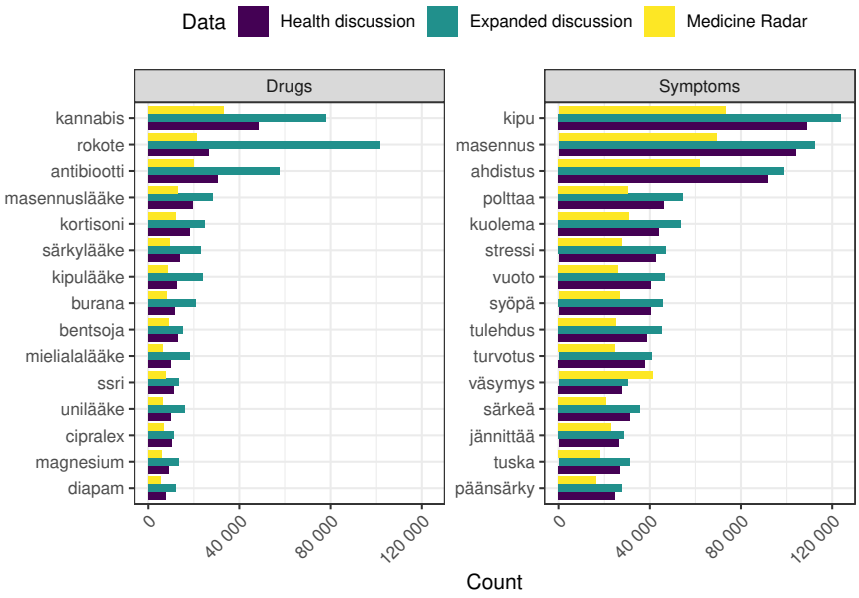

Figure 4.2: The most common drug and symptom words calculated from the health discussion messages

Comparing two word vector similarity models is not very straightforward, since there is no ground truth model that captures every word accurately. Model performance can not be evaluated using metrics like precision, recall or accuracy, because text retrieval is a human-centered process, which makes performance evaluation

Table 4.2: Translations of the most common drug and symptom words from Finnish to English

Drugs		Symptoms	
Finnish	English	Finnish	English
kannabis	*cannabis*	kipu	*pain*
rokote	*innoculation*	masennus	*depression*
antibiootti	*antibiotics*	ahdistus	*anxiety*
masennuslääke	*depression medication*	polttaa	*burn*
kortisoni	*cortisone*	kuolema	*death*
särkylääke	*pain medication*	stressi	*stress*
kipulääke	*pain medication*	vuoto	*leakage*
burana	*particular pain medication brand*	syöpä	*cancer*
bentsoja	*bentsodiatsepines*	tulehdus	*inflammation*
mielialalääke	*antidepressant*	turvotus	*swelling*
ssri	*SSRI*	väsymys	*tiredness*
unilääke	*hypnotic*	särkeä	*ache*
cipralex	*particular depression medication brand*	jännittää	*be nervous*
magnesium	*magnesium*	tuska	*pain*
diapam	*diazepam*	päänsärky	*headache*

difficult. It would require classification by human experts, who can identify and classify items as relevant or non relevant to rigorously evaluate performance of models. In this thesis, using such experts to classify drug or symptoms words was not possible. Ideally a list of word semantic relations, where items are representing the key concepts in a certain domain would be a suitable solution. Word2Vec models could be evaluated using intrinsic evaluation by comparing the expected results with the result produced by the model. In our knowledge this kind of list did not exist in Finnish language and producing such a list would require expertise in pharmacology and linguistics, which means that we had to utilize other existing resources. Both models produce 100 dimensional vectors for each word in their vocabulary, but we cannot make any conclusions based on the values, i.e. compare the value in the same dimension in two different vector spaces. However we can visualize the vector as a heatmap. The 15 most frequent drugs and symptoms are visualized in Figure 4.3. In this figure, the color represents the value Word2Vec has mapped for each word embedding. We can notice that the majority of dimensions include similar values. In Figure 4.3, the host prominent remark is the word "magnesium" has the most deviant values in some dimensions in both models. This is most probably due to the fact that magnesium can be interpreted as the element used in various other purposes, but also in medicinal use. From symptom words, we can observe that word "vuoto" (leakage, drain) also deviates from other words, because it has a different meaning in other contexts. Unexpectedly, its word embedding seems more similar with other symptom words in Model 2 than in Model 1.

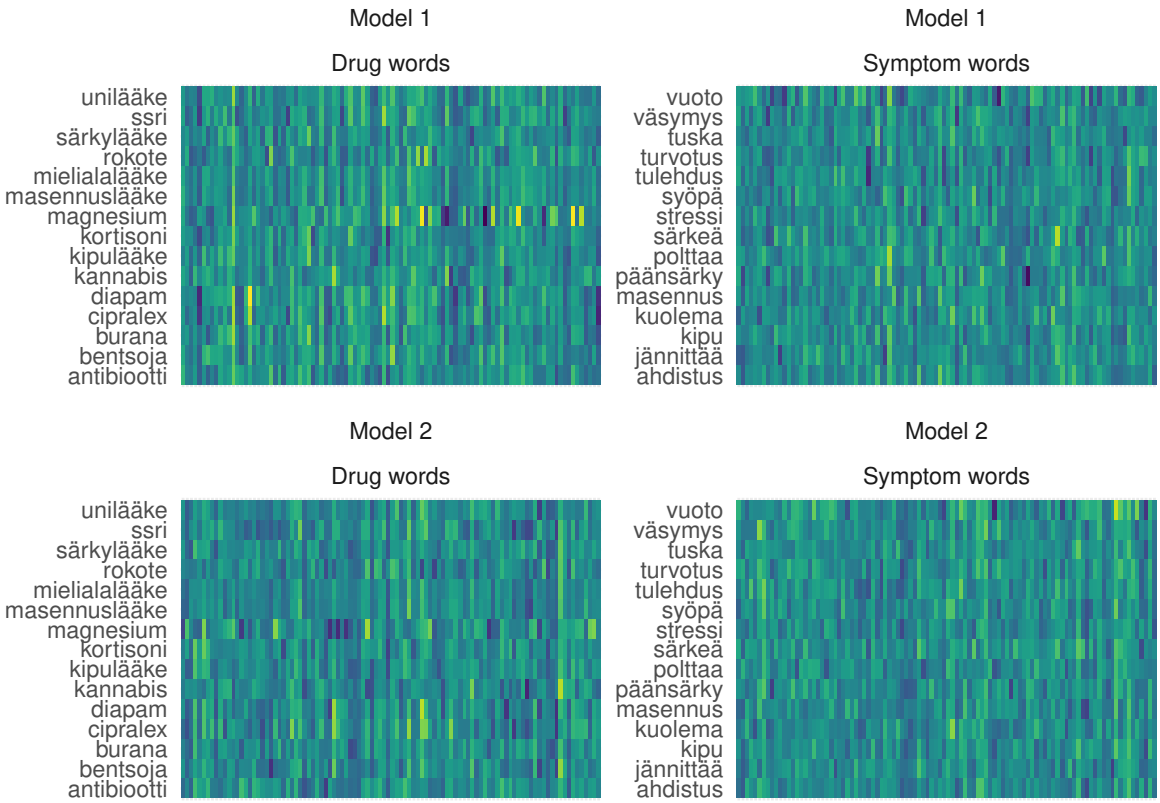

Figure 4.3: Visualizing word vector representations of the 15 most common drug and symptom words

4.2 Word Analogy Tasks

The most often used evaluation method of Word2Vec models is solving analogy tasks, suchs as described in Section 2.3. In our knowledge, there exists no such dataset of Finnish related words and creating such dataset would require linguistic and pharmacological expertise, which means that without such classified data, we can not measure the model performance or accuracy in solving analogy tasks involving drug and symptom words. In spite of the lack of sufficient resources, we can explore some of these analogies. Following the country capital example in Section 2.3, we have two positive examples of a symptom, “särky” (ache), and a a drug, “Burana” and a negative example of a symptom word,”masennus" (depression). A good result would be to receive a word that can be interpreted as a medicine related to depression. Using this examples, Model 1 suggest the most similar, i.e. shortest distance in vector space, to this analogy word vector representation is “särkylääke” (painkiller) and Model 2 suggest “panadol”, which is a marketing name for paracetamol.

Continuing with this test setting, we give two positive examples of one drug and one symptom word and the negative example word alternates between one drug

Table 4.3: Results of solving analogy tasks

Positive example 1		Positive example 2		Negative example		Model 1		Model 2	
Solve drug word analogy task									
särky	*ache*	burana	*burana*	masennus	*depression*	**särkylääke**	*painkiller*	**panadol**	*panadol*
syöpä	*cancer*	sytostaatti	*cytostatic drug*	allergia	*allergy*	**sädehoito**	*radiation therapy*	**sädehoito**	*radiation therapy*
ahdistus	*anxiety*	anksilon	*anksilon*	paniikki	*panic*	**voxra**	*voxra*	**voxra**	*voxra*
Solve symptom word analogy task									
zyrtec	*zyrtec*	allergia	*allergy*	sytostaatti	*cytostatic drug*	**allergialääke**	*allergy medicine*	**allergiaoire**	*allergy symptom*
alprox	*alprox*	paniikki	*panic*	anksilon	*anksilon*	**paniikkikohtaus**	*panic attack*	**pelkotila**	*state of fear*
prozac	*prozac*	masennus	*depression*	särky	*ache*	**depressio**	*depression*	**ssrilääkkeet**	*SSRIs*

and one symptom word. A good result in this test is to receive a similar drug-symptom word combination. The results in Table 4.3 indicate that in this task setting neither of the models do not perform very well. Both models suggest in all the example cases a word related to the positive examples, rather than the negative example. Based on these results, it seems that at least these health related concepts are mapped very close to each other in both vector spaces. It could be so that health words are in a cluster of their own in vector spaces, but based on this small experiment we can not conclude the effects of corpus size in this scenario.

4.3 Word Vector Space

Using t-distributed stochastic neighbor embedding (t-SNE) (Maaten et al., 2008) we can reduce the dimensions from 100 to 2 and plot some of the word embeddings in two dimensional planes. In Figure 4.4, on the left we have the 10 most similar words to “kannabis” (cannabis) and on the right the 10 most similar words to “kipu” (pain). Both input words are the most frequent words in the data with lots of training examples.

Use of cannabis is not legal in Finland neither in medical nor recreational use. Model 1 shows that similar words include alternative or colloquial names (“marihuana”, “ganjan”, "hamppu), but also different spellings both misspelled and different grammatical cases. Other words refer to illegal substances and recreational, but also some of the words indicate discussion around medicinal use.

For symptom words, most of the words are substantives or verb related different kinds of pain, but one of the words, “kivuta” is a lemma meaning the word climbing, which is not related to pain. Figure 4.5 represents the most similar word to both same input words, but in the Model 2, which was trained using the expanded data. On the left we see that now the most similar are strongly related to recreational use and only one word can be undoubtedly interpreted as related to medicinal use. The most probable explanation is that recreational cannabis use is more prevalent topic in other discussion areas as well and therefore the word embeddings are influenced by the introduction of these messages as training examples. For words similar to pain nonetheless are still different kind of pain. The

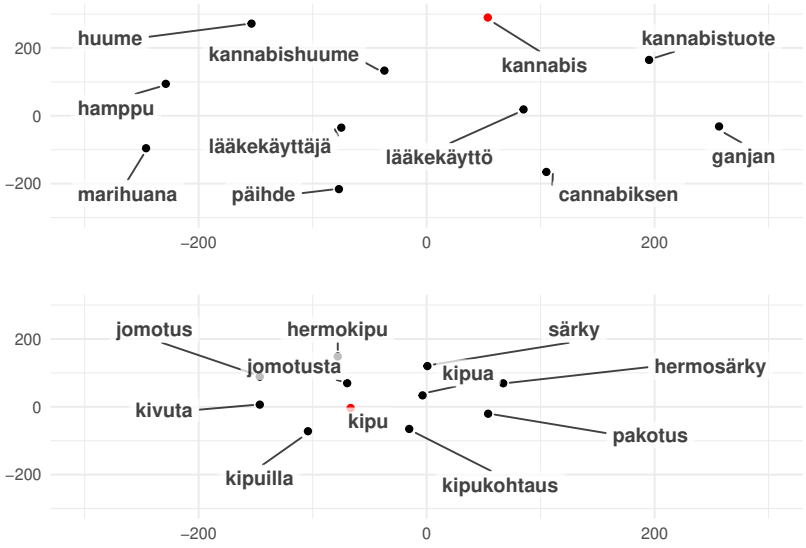

Figure 4.4: Most similar drug and symptom words in original vector space

word meaning climbing, “kivuta”, is still present among the 10 most similar words. One interesting observation is that Model 2 includes two words, “kipulääkitys” and “kipulääke”, which refer to pain medication. This result would be something that could be expected to be more prevalent in especially health discussion, but the inclusion of new messages clearly indicate the contrary.

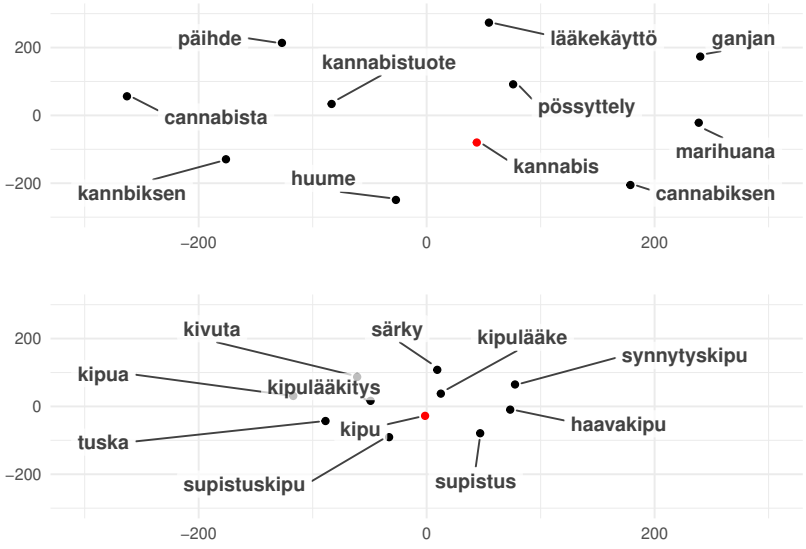

Figure 4.5: Most similar drug and symptom words in expanded vector space

It is easy to be fooled by the easiness to draw conclusions between the distances between two word embeddings, but it is important to remember that these plots were created from high-dimensional data originally, where the distance should be measured using measure appropriate for vector spaces, i.e. cosine similarity.

4.4 Evaluation of Cosine Similarities

The most important measure in word similarity models is cosine similarity (Equation (2.6)), because it measures the similarity between two vectors of an inner product space. Using the cosine of the angle between two vectors, it determines whether two vectors are pointing in roughly the same direction. We use cosine similarity to evaluate the effect of corpus size by computing the distances between two words in both models. We create three lists of word pairs using the concept vocabularies. We classify each concept either as a drug or a symptom word and create three comparison groups. First group includes only a pair of two drug words, the second group includes a pair of two symptom words and the third group includes one drug and one symptom word. The measured cosine similarities in these groups are visualized in Figure 4.6. Median of measures does not change drastically in any of the groups when trained using additional data. We can also see that the highest cosine similarity does not grow in any of the groups, but we see that the lowest measured cosine similarity is larger in all of the groups. There is no clear indication of improvement of capturing semantic similarity except in the groups of two drug words where the median is a bit higher and the quartile limits grow smaller meaning that most of measured similarities are between 0.50-0.75. Relationships between the cosine similarity measured from Model 1 and the co-

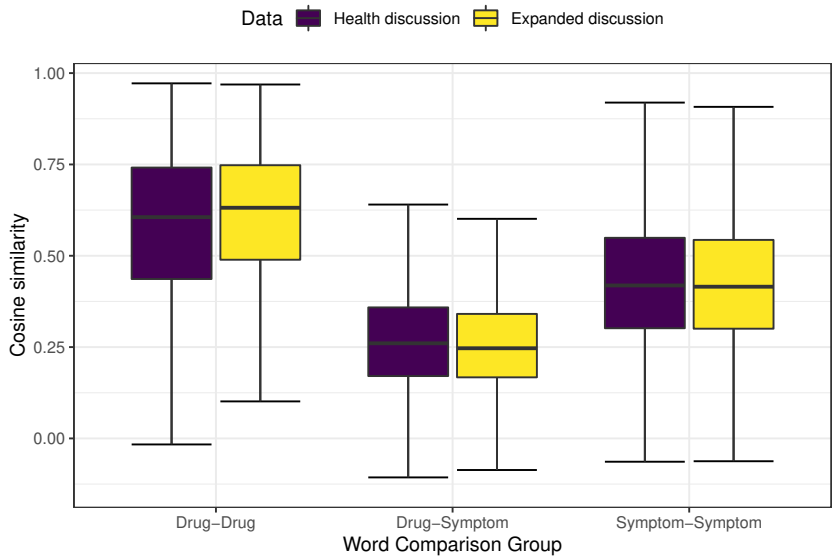

Figure 4.6: Box plot of measured cosine similarity

sine similarity measured from Model 2 are represented in Figure 4.7. In this figure, we expect a result where the point falls in the $x = y$ diagonal line. For example in the leftmost picture many of word pairs with a cosine similarity between 0.50-0.75 has measured much lower similarity score in Model 2, which can be caused by noise in the expanded data. On the other hand, the group of only symptom words show that both of the models measured the same value of cosine similarity. The group one drug and one symptom word seems to have very mixed results as there is some deviation, but also the measured similarities are much smaller in general than in the two other groups.

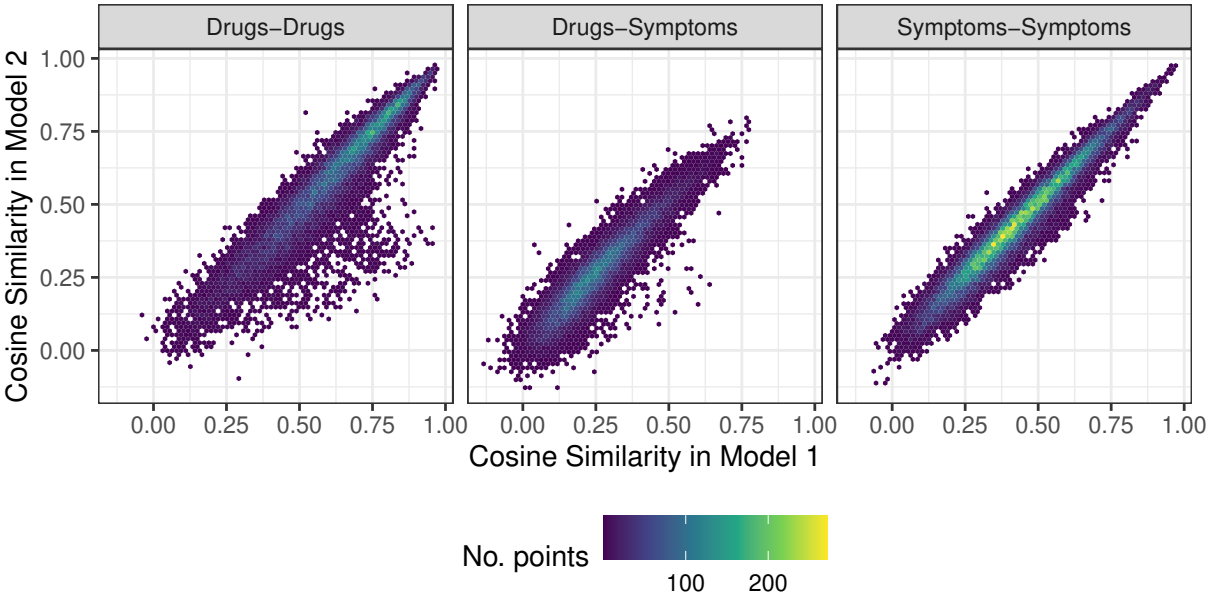

Figure 4.7: Cosine similarities of word pairs in both models

Chapter 5

Conclusions

The results show that no clear connection exists between corpus size and the measured cosine similarity distances in two vector spaces. Also increasing the size of data by almost 30% did not have an effect on the characteristics of training data, but further researching the differences of the datasets would require more detailed qualitative text analytics. The level of noise in data was unexpectedly very minimal. Adding training examples to domain specific corpus from other discussion areas did not have an explicit effect on cosine similarities.

We observed that drug words were more sensitive for the effect of adding data. This is most probably due to the fact that most of the expanded messages come from societal discussion and that also illegal substances were included in the Medicine Radar project which provided the basis of concept vocabularies. The effect of new training samples was less prevalent in results involving symptom words as measured cosine similarities were reasonably as they were with the cosine similarities measured from the health discussion. The combined group of a single drug and a single symptom word provided mixed results. Results of this group were affected by the results of both of the other groups as pairs of drug words were more influenced than the pairs of symptom words.

Drug names are also very distinguishable and unambiguous when compared to symptoms. This means that context can be misinterpreted when dealing with symptoms more often than when observing words representing drugs. Naturally the symptom words can appear in everyday conversations that are not related to health specifically.

To conclude, it seems that the most prominent effect of corpus size was that the health words seem to be in a cluster of their own separate from the other words included in training examples. Word2Vec does not seem to be able to differentiate drugs and concepts itself, but the expanded training data included discussion less favorable to drugs in terms of learning the semantic similarity.

Chapter 6

Discussion

There is clearly a need for better Finnish language resources to evaluate natural language processing tasks. Detailed language resources could improve evaluation in a similar text mining framework. Full potential of Word2Vec or the concept vocabularies of Medicine Radar was not utilized. Problems persisted when combining the method to produce concept vocabulary with word similarity model training. In most cases, the medicinal words in Suomi24 data were not lemmatized properly. This is most probably due to Finnish Dependency Parser not recognizing the word in order to process it properly. Often the captured expressions included long compound words, which are very common in Finnish.

The word lists of the Medicine Radar project are the most applicable resources available for text mining Finnish health related non-medicinal texts. The same concept vocabularies produced in the project can be applied so that we can utilize the same lists to catch the different expressions and spelling from other datasets and from other domains as well.

Using external domain expertise from other fields of research would provide support to develop also methods to evaluate the performance and accuracy of word similarity models. Creating analogy datasets of very domain specific word analogies would enable a more simple way to evaluate the semantic similarities between words represented by their vectors.

The future research could also focus on different metrics to compare the measured similarities, methods to rank the relationships between two words or to measure the strength or statistical power of the relationship. Also the setting used in this thesis could be reversed by studying the effects of corpus size by decreasing the number of training examples.

Word similarity models representing domain specific and relevant concepts provide a great opportunity to combine quantitative methods with qualitative research. That would imply studying both the methodology behind producing the word embeddings as well as analyzing the concepts and what they represent.

References

Autio, M. M., Helovuori, S., & Autio, J. (2012). Potilaskuluttajan ja lääkärin muuttuvat roolit sähköistyvillä terveysmarkkinoilla. *Kulutustutkimus.Nyt : Kulutustutkimuksen Seuran Julkaisu.*, *6*(2), 40–57.

Bengio, Y., Ducharme, R., Vincent, P., & Janvin, C. (2003). A neural probabilistic language model. *Journal of Machine Learning Research*, *3*, 1137–1155.

Chen, L., Yuan, F., Jose, J. M., & Zhang, W. (2018). Improving negative sampling for word representation using self-embedded features. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining* (pp. 99–107).

Chen, S. F., & Goodman, J. (1999). An Empirical Study of Smoothing Techniques for Language Modeling. *Computer Speech & Language*, *13*(4), 359–393.

Chiu, B., Crichton, G., Korhonen, A., & Pyysalo, S. (2016). How to Train good Word Embeddings for Biomedical NLP. In *Proceedings of the 15th Workshop on Biomedical Natural Language Processing* (pp. 166–174).

Goldberg, Y. (2017). *Neural network methods for natural language processing* (Vol. 37). Morgan & Claypool.

Hardey, M. (2001). 'E-health': The internet and the transformation of patients into consumers and producers of health knowledge. *Information, Communication & Society*, *4*(3), 388–405.

Jurafsky, D., & Martin, J. H. (2008). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition* (Vol. 2). Upper Saddle River, New Jersey: Pearson Prentice Hall.

Kohonen, T. (2001). *Self-organizing maps* (Third edition.). Berlin: Springer.

Kohonen, T., Oja, E., Simula, O., Visa, A., & Kangas, J. (1996). Engineering applications of the self-organizing map. *Proceedings of the IEEE*, *84*(10), 1358–1384.

Lagus, K. (2000). *Text mining with the websom*. Helsinki University of Technology.

Lagus, K., Pantzar, M., & Ruckenstein, M. (2015). Keskustelun tunneaallot – Suomi24-hanke. *Tieteessä Tapahtuu*, *33*(6), 39–41.

Lagus, K., Ruckenstein, M., Juvonen, A., & Rajani, C. (2018). Medicine radar–a tool for exploring online health discussions. In *Proceedings of the digital humanities in the nordic countries 3rd conference*. CEUR-WS. org.

Lagus, K., Ruckenstein, M., Pantzar, M., & Ylisiurua, M. (2016). *Suomi24: Muodonantoa aineistolle*. University of Helsinki.

Lai, S., Liu, K., He, S., & Zhao, J. (2016). How to Generate a Good Word Embedding? *IEEE Intelligent Systems*, *31*(6), 5–14.

Lison, P., & Kutuzov, A. (2017). Redefining context windows for word embedding models: An experimental study. Retrieved from `http://arxiv.org/abs/1704.05781`

Maaten, L. van der, & Hinton, G. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research*, *9*(Nov), 2579–2605.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. Retrieved from `http://arxiv.org/abs/1301.3781`

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems* (Vol. 26, pp. 3111–3119).

Mnih, A., & Hinton, G. E. (2008). A scalable hierarchical distributed language model. In *Advances in neural information processing systems* (Vol. 21, pp. 1081–1088).

Morin, F., & Bengio, Y. (2005). Hierarchical Probabilistic Neural Network Language Model. In *Aistats* (Vol. 5, p. 6).

Rong, X. (2016). Word2vec Parameter Learning Explained. *arXiv:1411.2738 [Cs]*.

Th, M., Sahu, S., & Anand, A. (2015). Evaluating distributed word representations for capturing semantics of biomedical concepts. In *Proceedings of BioNLP 15* (pp. 158–163).

Turney, P. D., & Pantel, P. (2010). From frequency to meaning: Vector space models of semantics. *The Journal of Artificial Intelligence Research*, *37*, 141–188.

Ylisiurua, M. (2017). Aihemallinnuksen mahdollisuudet sosiaalisen median aineistojen jäsentämisessä: Terveyskeskustelu suomi24-verkkopalstalla. *Kulutustutkimus. Nyt: Kulutustutkimuksen Seuran Julkaisu*, *11*(2), 44–67.